## PROGRAMA3CIÓN ORIENTADA A OBJETOS

**Excepciones Octubre 2017 Laboratorio 4/6**

**OBJETIVOS**

** Perfeccionar el diseño y código de un proyecto considerando casos especiales y errores.

** Construir clases de excepción encapsulando mensajes.

** Manejar excepciones considerando los diferentes tipos.

** Registrar la información de errores que debe conocer el equipo de desarrollo de una aplicación en producción.

** Vivenciar la prácticas ***Designing*** [*Simplicity*](http://www.extremeprogramming.org/rules/simple.html)*.*

**Coding** Code must be written to agreed [standards](http://www.extremeprogramming.org/rules/standards.html)

## ENTREGA

 Incluyan en un archivo .zip los archivos correspondientes al laboratorio. El nombre debe ser los dos apellidos de los miembros del equipo ordenados alfabéticamente.

 Deben publicar el avance al final de la sesión y la versión definitiva en la fecha indicada,

en los espacios preparados para tal fin.

# EQUIPOS EN BLUEJ

**PRACTICANDO MDD y BDD con EXCEPCIONES [En lab04.doc, equipos.asta y BlueJ equipos**

En este punto vamos a aprender a diseñar, codificar y probar usando excepciones. Para esto se van a trabajar dos métodos de la clase Equipo y la excepción EquipoExcepcion

1. En su directorio descarguen los archivos contenidos en equipo.zip, revisen el contenido y estudien el diseño estructural de la aplicación.
2. ¿De qué clase no pueden consultar las fuentes? ¿En qué formato está? ¿Por qué? Estudien su documentación en el diseño o en el código (/doc).

De la clase “Persona”, esta en formato documentación

1. Dadas las pruebas, diseñen y codifiquen el método valorHora.
2. Dada la especificación, diseñen, codifiquen y prueben el método valorHoraEstimado.

### PARA LAS PRUEBAS

Las siguientes personas que tienen valor hora Las siguientes personas son conocidas pero no conocido: tienen valor hora:

("Pedro",10000); ("Garcia");

("Santiago",20000); ("Ospina");

("Marcos",30000); ("Guarin")

("Juan",40000);

("Judas",50000);

**SINAP. Sistema Nacional de Áreas Protegidas**

**EN CONSOLA**

**Conociendo el proyecto SINAP [En lab04.doc]**

**No olviden respetar los directorios bin docs src**

1. En su directorio descarguen los archivos contenidos en SINAP.zip, revisen el contenido.

¿Cuántos archivos se tienen? ¿Cómo están organizados? ¿Cómo deberían estar organizados?

5 archivos , 4 que corresponden a clases y un archivo astah, los archivos .java deberian estar dentro de una carpeta src

1. Estudien el diseño del programa: diagramas de paquetes y de clases.

¿cuántos paquetes tenemos?

Hay dos paquetes , aplicación y presentacion

¿cuántas clases tiene el sistema?, ¿cómo están organizadas?

Presentacion tiene una clase SinapGUI

Aplicación tiene3 clases Area,Registro,Sinap

4 clases en total

¿cuál es la clase ejecutiva?

La clase ejecutiva es Sinap GUI , porque es la clase que utiliza los dos paquetes

1. Prepare los directorios necesarios para ejecutar el proyecto. ¿qué estructura debe tener?

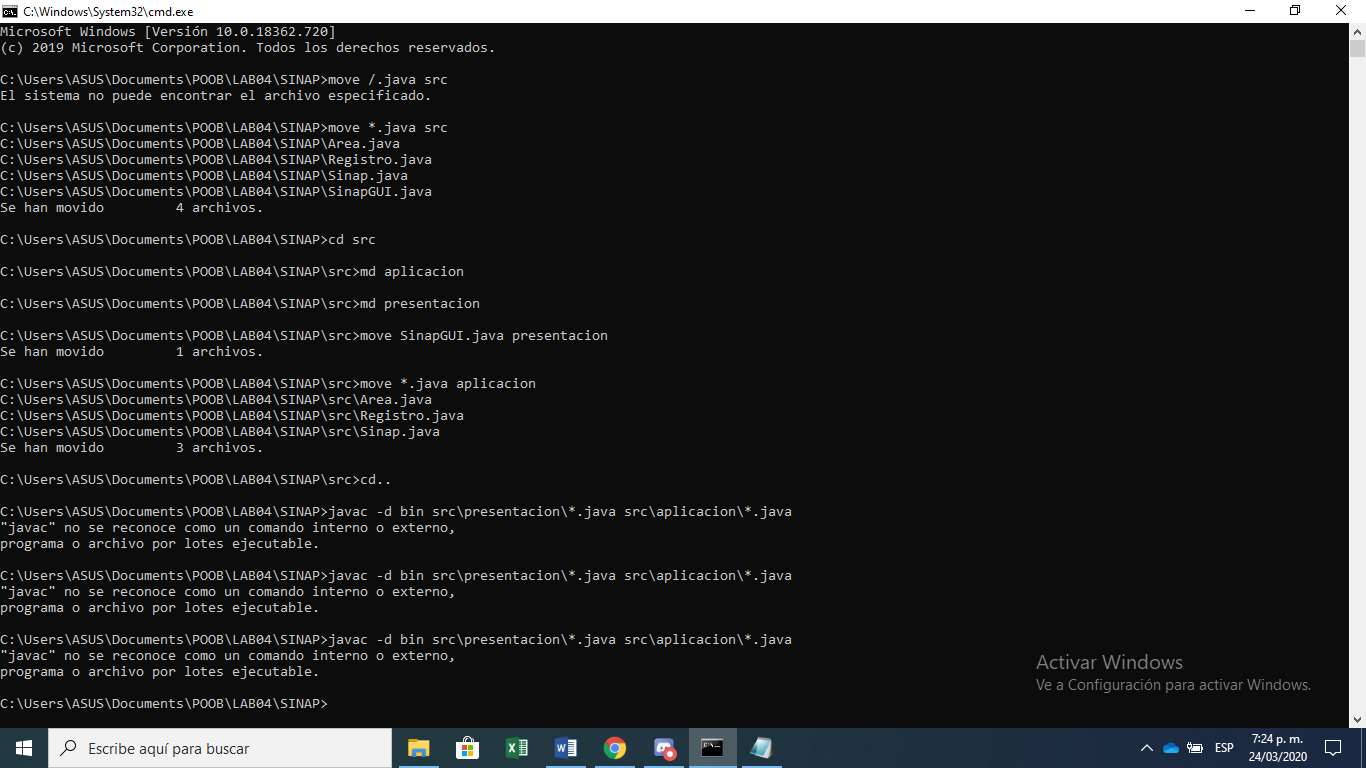
¿qué instrucciones debe dar para ejecutarlo?

Deberia tener esta estructura

SINAP

src

bin

 docs

javac -d bin src\presentacion\\*.java src\aplicacion\\*.java

java -cp bin presentacion.SinapGUI

1. Ejecute el proyecto, ¿qué funcionalidades ofrece? ¿cuáles funcionan?

Ofrece las funcionalidades listar,adicionar y buscar

1. Realicen el diagrama de casos de uso correspondiente.
2. ¿De dónde salen las áreas iniciales? Revisen el código y la documentación del proyecto.

¿Qué clase pide que se adicionen? ¿Qué clase los adiciona?

En la clase SINAP el método “adicioneCinco” , las adiciona

**Adicionar y listar. Todo OK.** [En lab04.doc, SINAP.asta y \*.java]

(NO OLVIDEN BDD - MDD)

El objetivo es realizar ingeniería reversa a las funciones de adicionar y listar.

1. Adicionen una nueva área

Tuparro

Tuparro National Park Vichada

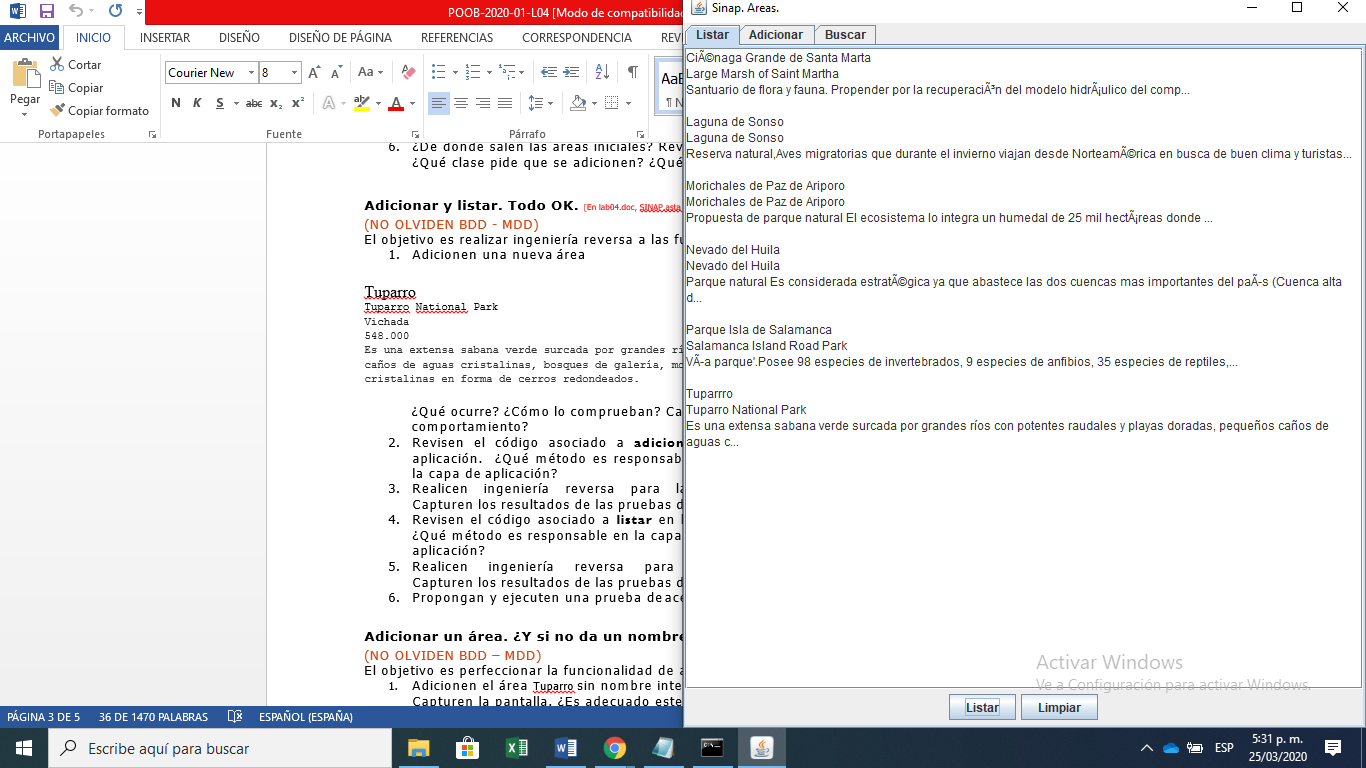
548.000

Es una extensa sabana verde surcada por grandes ríos con potentes raudales y playas doradas, pequeños caños de aguas cristalinas, bosques de galería, morichales y saladillales, además de enormes rocas cristalinas en forma de cerros redondeados.

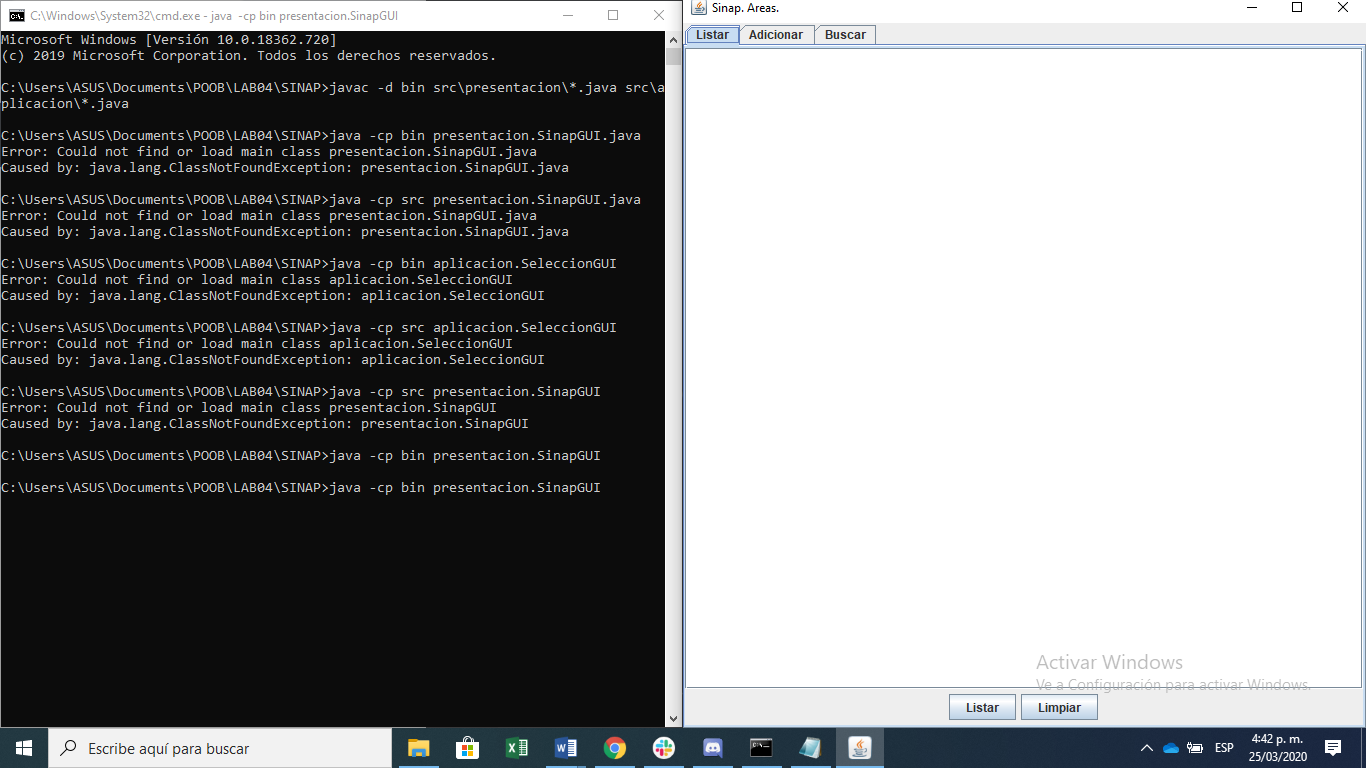
¿Qué ocurre?

Lo adiciona pide los datos del nombre, nombre internacional, ubicación, área y descripción.

¿Cómo lo comprueban? Capturen la pantalla.

Lo comprobamos dándole en la parte de listar y ahí aparece el elemento que adicionamos

¿Es adecuado este comportamiento?

Si, por que cumple con lo que esta implementado.

1. Revisen el código asociado a **adicionar** en la capa de presentación y la capa de aplicación.

¿Qué método es responsable en la capa de presentación?

PrepareAreaAdicionar

¿Qué método en la capa de aplicación?

AdicionaDetalles

1. Realicen ingeniería reversa para la capa de aplicación para **adicionar**. Capturen los resultados de las pruebas de unidad.

@Test

deberiaAdicionar()

1. Revisen el código asociado a **listar** en la capa de presentación y la capa de aplicación.

¿Qué método es responsable en la capa de presentación?

AreaListar

¿Qué método en la capa de aplicación?

To String

1. Realicen ingeniería reversa para la capa de aplicación para **listar**. Capturen los resultados de las pruebas de unidad.

@Test

deberiaListar()

1. Propongan y ejecuten una prueba de aceptación.

@Test

pruebaDeUnidad()

**Adicionar un área. ¿Y si no da un nombre?** [En lab04.doc, SINAP.asta y \*.java]

(NO OLVIDEN BDD – MDD)

El objetivo es perfeccionar la funcionalidad de adicionar un área.

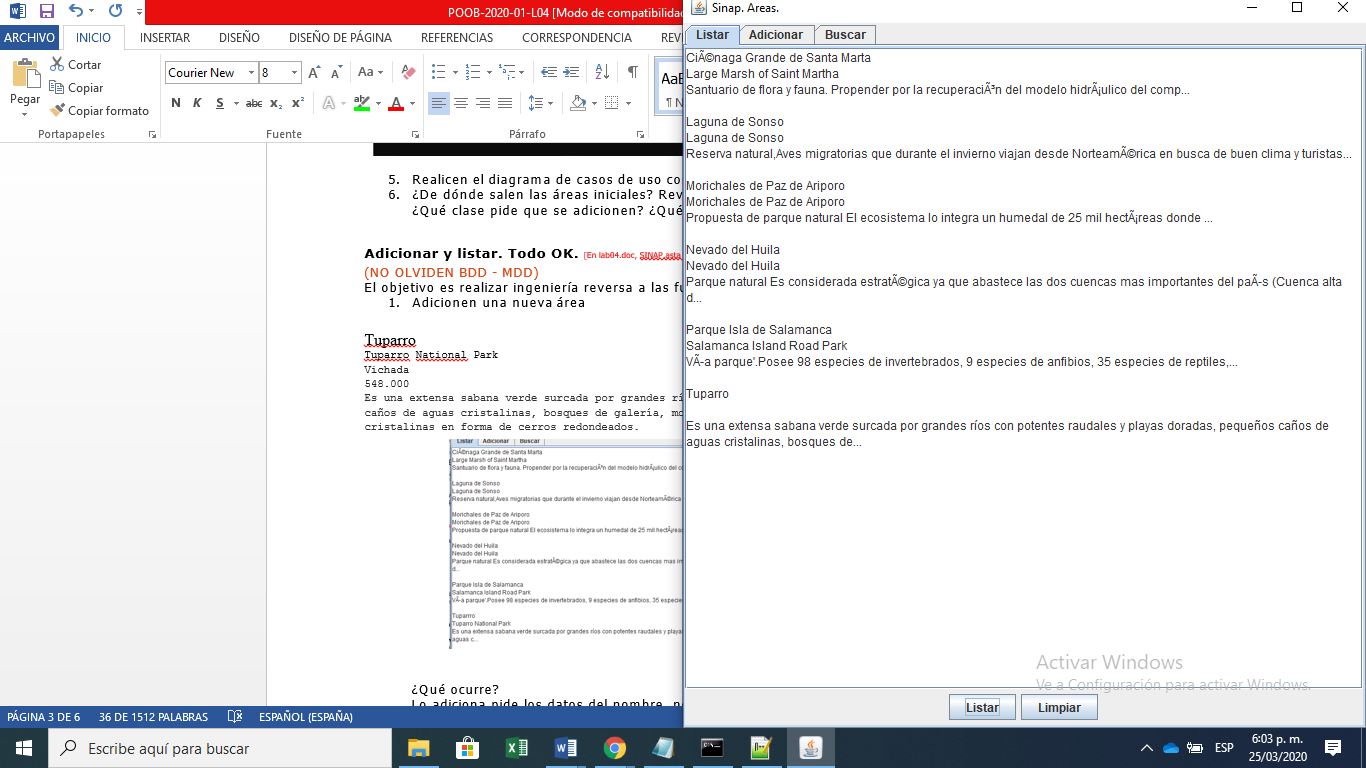
1. Adicionen el área Tuparro sin nombre internacional.

¿Qué ocurre?

Finalmente lo adiciona

¿Cómo lo comprueban?

Listandolo una vez mas

Capturen la pantalla.

¿Es adecuado este comportamiento?

No estoy de acuerdo con este comportamiento, ya que no debería dejarlo adicionar porque el nombre internacional puede ser un dato muy importante para el usuario.

1. Vamos a evitar la creación de áreas con un nombre internacional vacío manejando una excepción SINAPExcepcion. Si el área no tiene nombre internacional, no la creamos y

lo comunicamos al usuario[1](#_bookmark0). Para esto lo primero que debemos hacer es crear la nueva clase SINAPExcepcion considerando este primer mensaje.

1. Analicen el diseño realizado.
2. ¿Qué método debería lanzar la excepción?

Registre

¿Qué métodos deberían propagarla?

Adicione porque es el método que comprueba que el nombre internacional es vacío

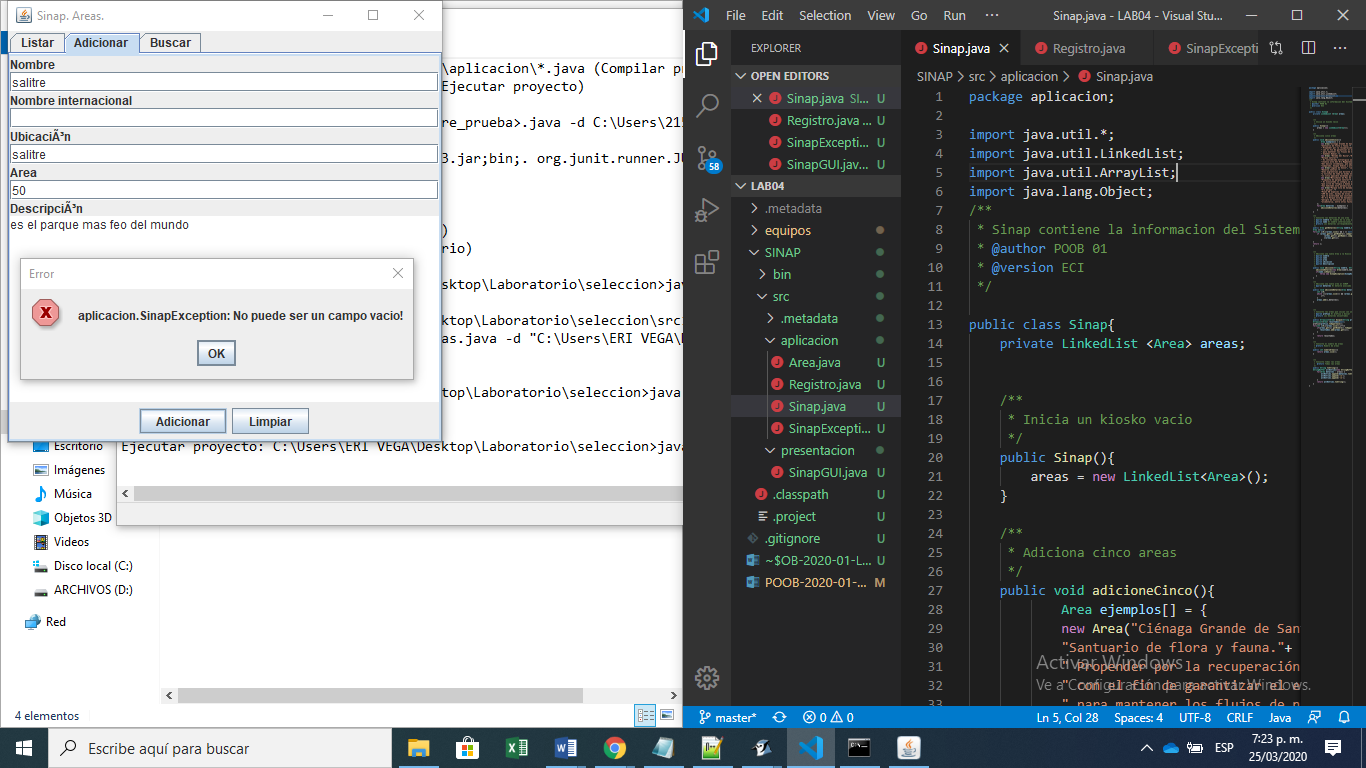
¿Qué método debería atenderla? Explique claramente.

El método accionAdicionar de SinapGUI debería atenderla

1. Construya la solución propuesta. Capturen los resultados de las pruebas.

@Test

deberiaFallarSinNombreInternacional

1. Ejecuten nuevamente la aplicación con el caso de prueba propuesto en 1.,

Capture la pantalla.

¿Qué sucede ahora?

Lanza la excepción.

**Adicionar un área. ¿Y si ya se encuentra?** [EnSINAP.asta, lab04.java y \*.java]

(NO OLVIDEN BDD - MDD)

El objetivo es perfeccionar la funcionalidad de adicionar un área.

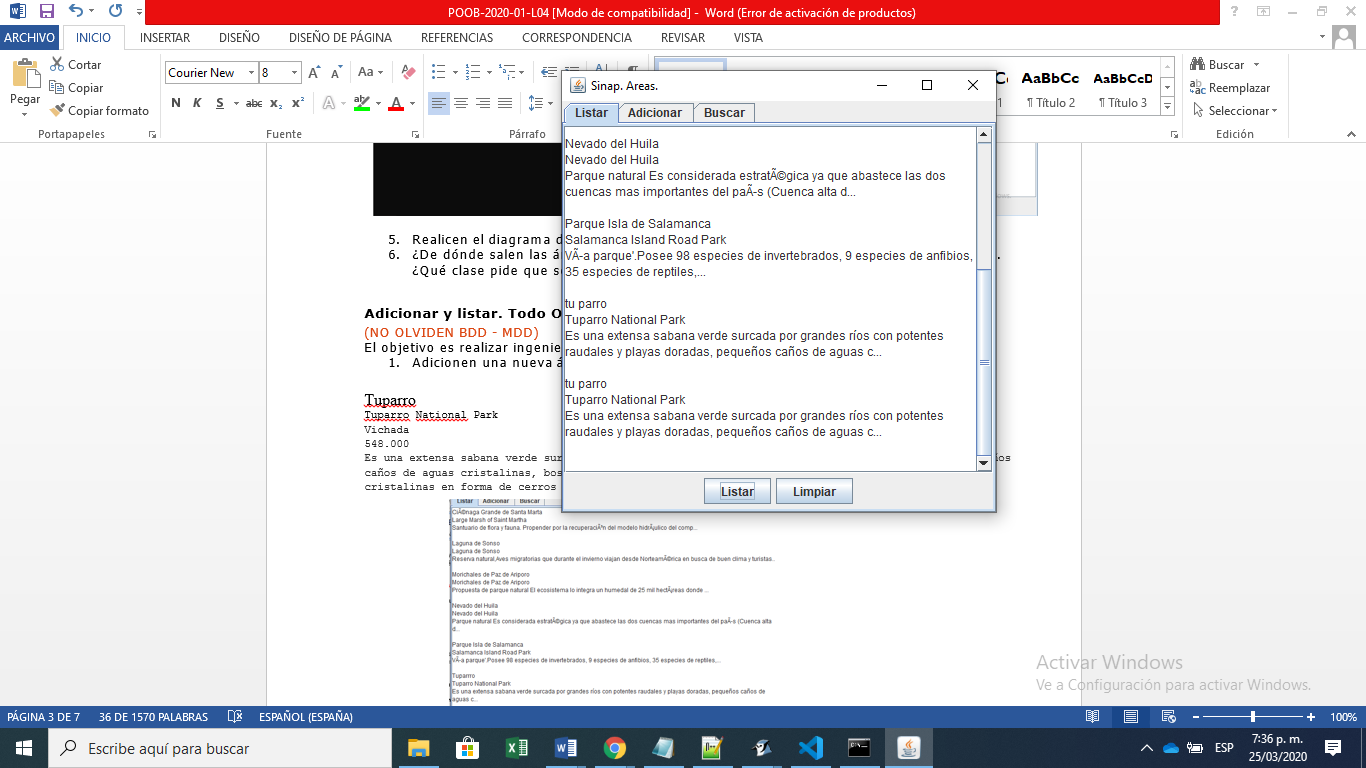
1. Adicionen dos veces la nueva área Tuparro

¿Qué ocurre?

Lo deja añadir

¿Cómo lo comprueban?

Listándolo

Capturen la pantalla.

¿Es adecuado este comportamiento?

No porque va a hacer repetido y no va servir de nada

1. Analicen el diseño realizado.

¿Qué método debería lanzar la excepción?

Registre de la clase registro

¿Qué métodos deberían propagarla?

Adicione porque es el método encargado de adicionar x cantidad de veces

¿Qué método debería atenderla? Explique claramente.

accionAdicionar de la clase SINAPGUI

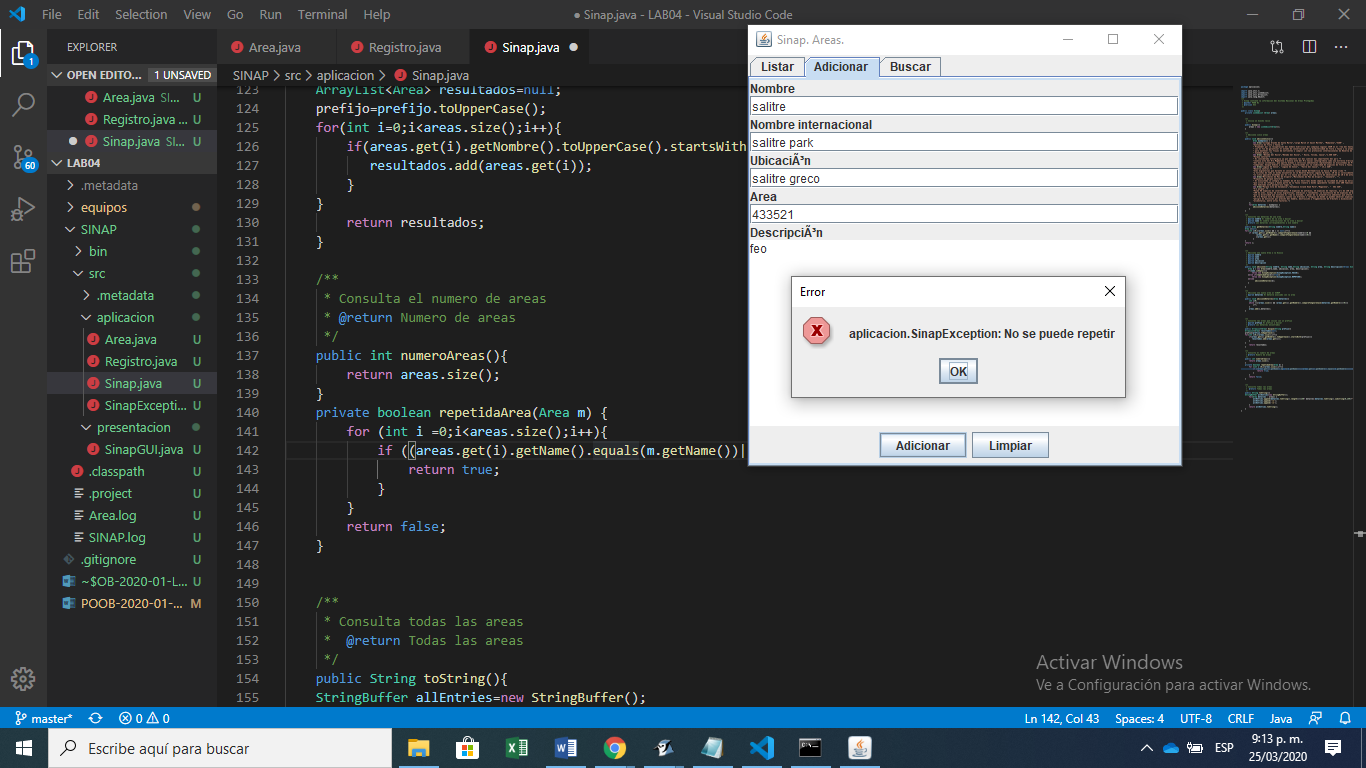
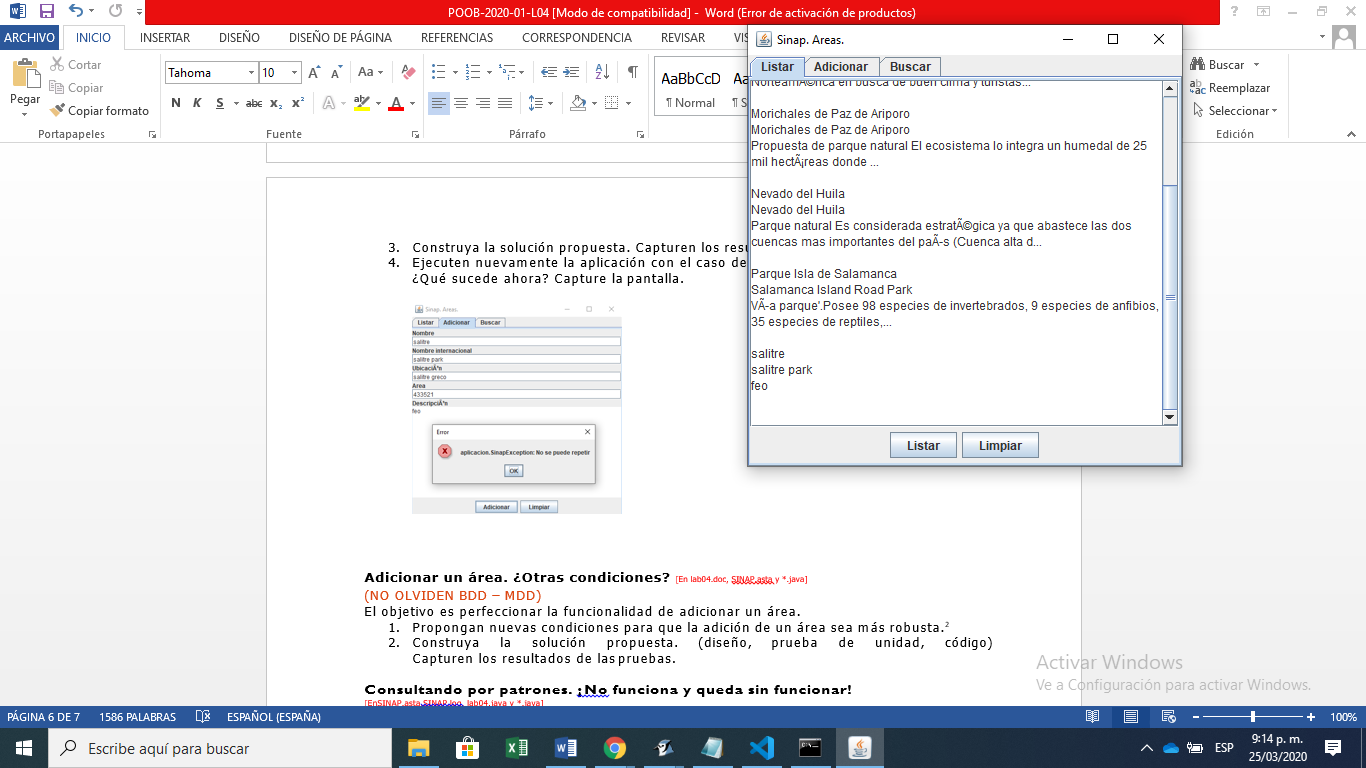
1. Construya la solución propuesta. Capturen los resultados de las pruebas.

@Test

deberiaFallarConAreaRepetida()

1. Ejecuten nuevamente la aplicación con el caso de prueba propuesto en 1.,

¿Qué sucede ahora? Capture la pantalla.

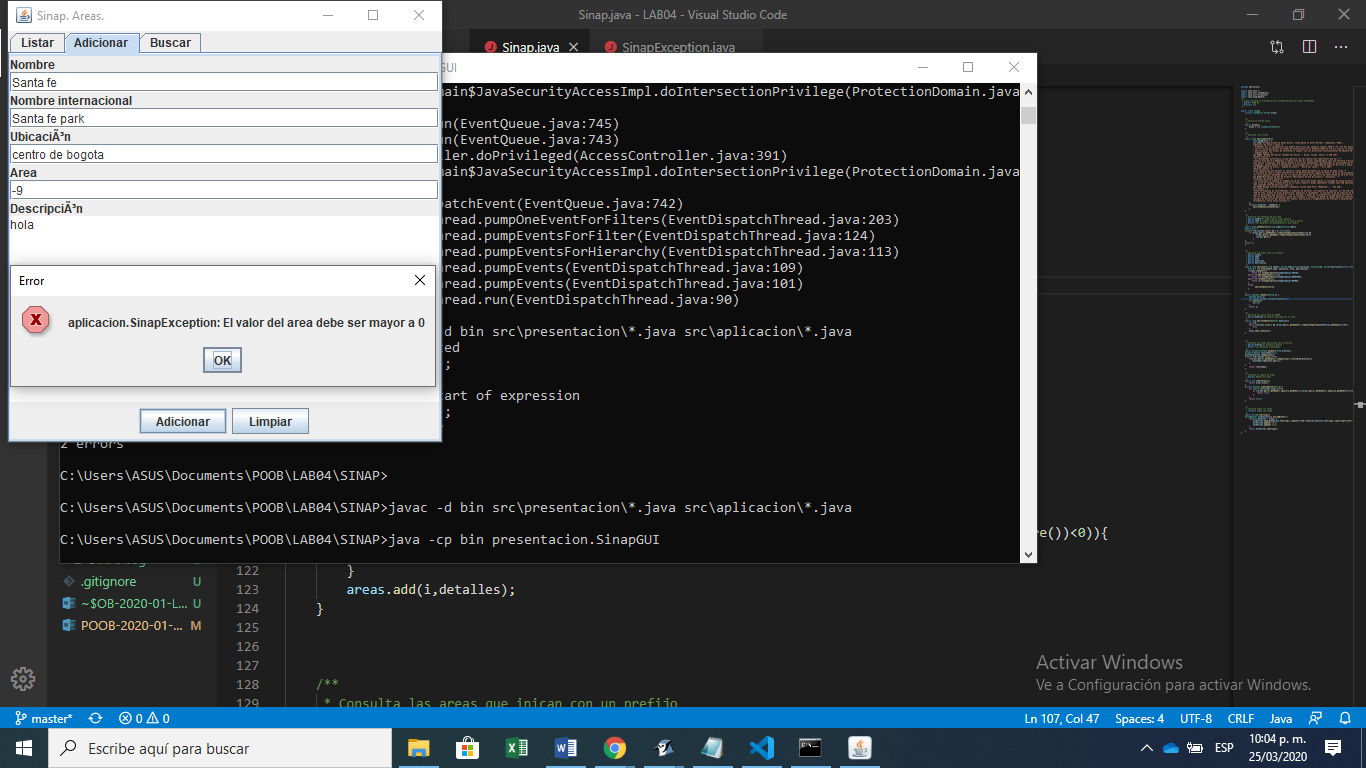


**Adicionar un área. ¿Otras condiciones?** [En lab04.doc, SINAP.asta y \*.java]

(NO objetivo es perfeccionar la funcionalidad de adicionar un área.

1. Propongan nuevas condiciones para que la adición de un área sea más robusta.[2](#_bookmark1)

Proponemos la condición de que área sea mayor a 0

1. Construya la solución propuesta. (diseño, prueba de unidad, código) Capturen los resultados de las pruebas.

@Test()

deberiaFallarConAreaMenorACero()

OLVIDEN BDD – MDD)

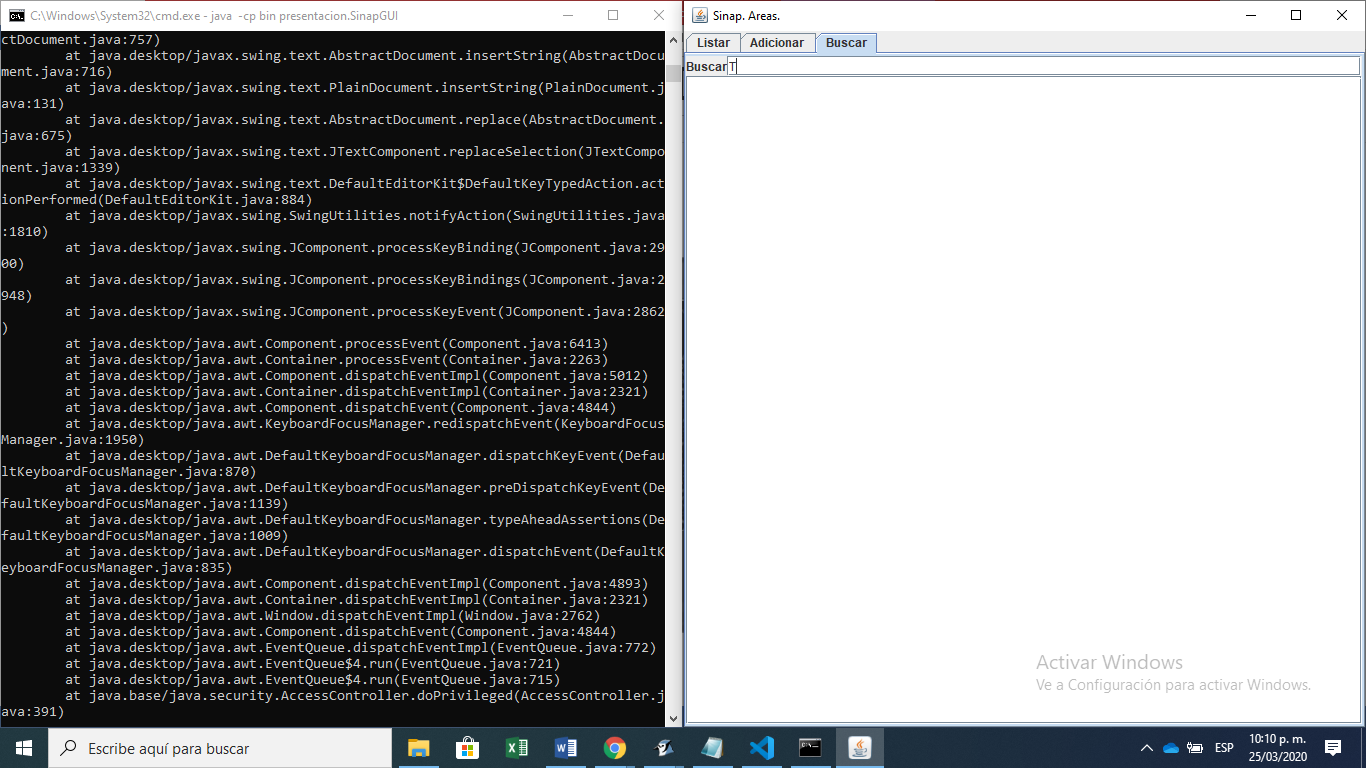
El

### Consultando por patrones. ¡ No funciona y queda sin funcionar!

[EnSINAP.asta,SINAP.log, lab04.java y \*.java]

(NO OLVIDEN BDD - MDD)

 Consulten un área especial que inicie con T. ¿Qué sucede? ¿Qué creen que pasó?

Capturen el resultado. ¿Quién debe conocer y quien NO debe conocer esta información?

Sucede que hay un error al buscar las area, el programador debe conocer los errores de la aplicación, pero el usuario NO.

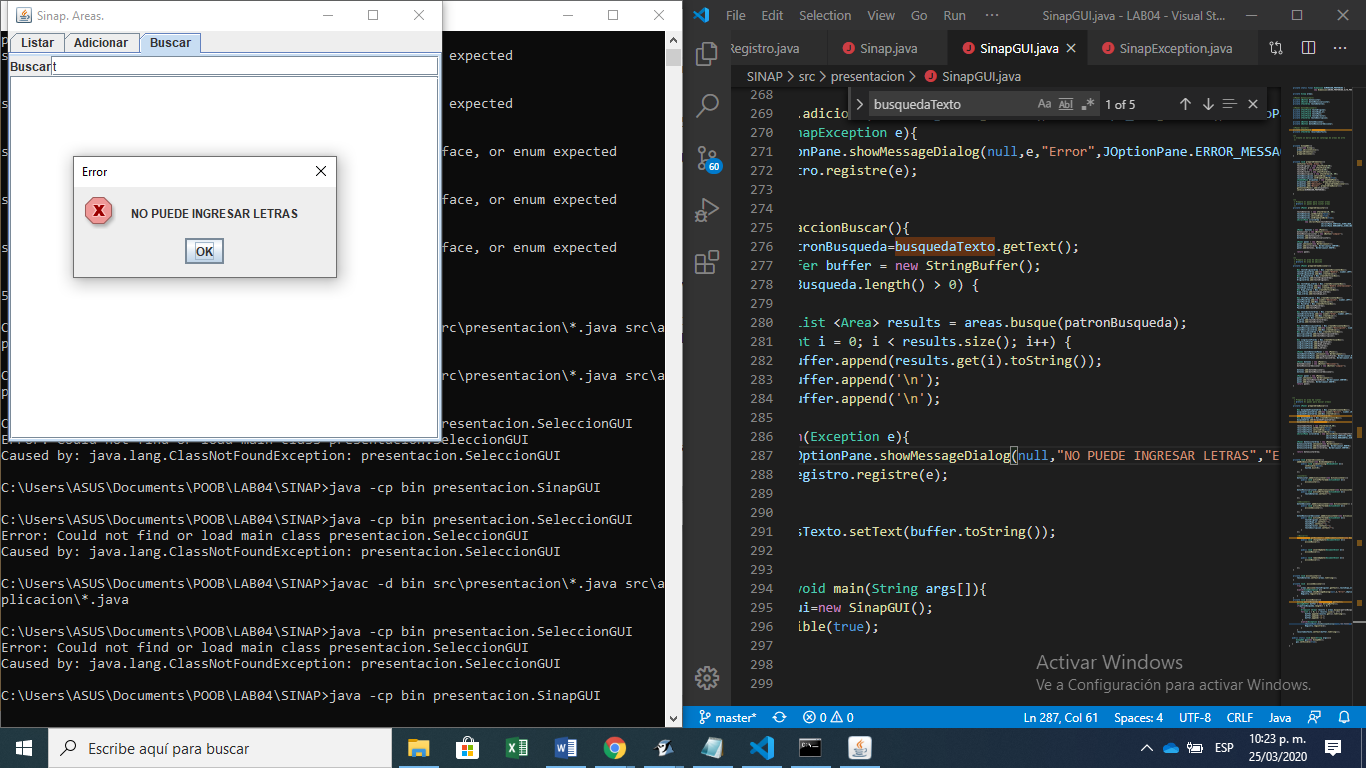
 Exploren el método registre de la clase Registro ¿Qué servicio presta?

El método registre se encarga de llevar una bitácora, en donde en caso de haber alguna excepción, nos va a mostrar el rastro de esta.

 Analicen el punto adecuado para que **SIEMPRE,** al sufrir en cualquier punto el sistema un incidente como este, se presente un mensaje especial de alerta al usuario, se guarde la información del error en el registro de error y termine la ejecución. Expliquen y construyan la solución.

Se realiza un try and catch de manera que si no es posible hacer una búsqueda, la aplicación sea capaz de poder tratar la excepción y poder continuar su ejecución, y para poder mostrar al usuario este error se muestra una alerta

 Ejecuten nuevamente la aplicación con el caso propuesto en 1. ¿Qué mensaje salió en pantalla? ¿La aplicación termina? ¿Qué información tiene el archivo de errores?

Se muestra la alerta definida anteriormente, la aplicación continua su ejecución.

 ¿Es adecuado que la aplicación continúe su ejecución después de sufrir un incidente como este? ¿de qué dependería continuar o parar?

Debería parar , si hay un error no debería seguir su ejecucion

1 Para presentar los mensajes de error al usuario use el méodo de clase de JOptionPane public static void **showMessageDialog**(Component parentComponent,

Object message, String title, int messageType)

throws HeadlessException

Con componente padre:este mensaje: la cadena correspondiente al mensaje de error de la excepcion correspondiente, titulo: ERROR y tipo de mensaje: JOptionPane.ERROR\_MESSAGE

2Robustez o solidez. Se refiere a la capacidad del software de defenderse de las acciones anormales que llevan al sistema a un estado no deseado o por lo menos no previsto, causando un comportamiento inesperado, indeseado y posiblemente erróneo

 Analicen el punto adecuado para que **EN ESTE CASO** se presente un mensaje especial de alerta al usuario, se guarde la información del error en el registro y continúe la ejecución. Expliquen y construyan la solución. No eliminen la solución de 3.

 Ejecuten nuevamente la aplicación con el caso propuesto en 1. ¿Qué mensaje salió en pantalla? ¿La aplicación termina? ¿Qué información tiene el archivo de errores?

### Consultando por patrones. ¡Ahora si funciona!

(NO OLVIDEN BDD - MDD)

1. Revisen el código asociado a **buscar** en la capa de presentación y la capa de aplicación.

¿Qué método es responsable en la capa de presentación? ¿Qué método es responsable en la capa de aplicación?

prepareAreaBuscar en SINAPGUI de la capa de presentación

toString de SINAP en la capa de aplicacion

1. Realicen ingeniería reversa para la capa de aplicación para **buscar**. Capturen los resultados de las pruebas. Deben fallar.

@Test

deberiaFallarBusque()

1. ¿Cuál es el error? Soluciónenlo. Capturen los resultados de las pruebas.
2. Ejecuten la aplicación nuevamente con el caso propuesto. ¿Qué tenemos en pantalla?

¿Qué información tiene el archivo de errores?

**RETROSPECTIVA**

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)

Lina 20

Santiago 20

1. ¿Cuál es el estado actual del laboratorio? ¿Por qué?

Parcialemente completo , todos los puntos estas completos , talvez faltan algunos detalles el ultimo punto, no era claro

1. Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?

Programacion a pares y pruebas de unidad, era necesario hacer varias pruebas

1. ¿Cuál consideran fue el mayor logro? ¿Por qué?

Lograr compilar y ejecutar el programa por consola, porque era un tema nuevo

1. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

Compilar , ejecutar por consola y aprender a leer errores por consola , ya que era un tema nuevo , lo resolvimos investigando y probando.

1. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Buena comunicación , mejor manejo del tiempo y compromiso.